

# SPRAWOZDANIE

Autor: Michał Deć

Opiekun pracy: **dr inż. prof. PRz Mariusz Borkowski**

Rzeszów, rok 2021

## 1. Wstęp

Program ma za zadanie wyszukanie największej możliwej liczby, konstruując ją poprzez złączenie podanych uprzednio liczb.

Przykład:

Wejście: [10, 81, 9]

Wyjście: 98110

## 2. Opis implementacji problemu i podstaw zagadnienia

Należy zauważyć, że aby stworzyć największą możliwą liczbę z uprzednio podanych liczb, to należy te liczby ustawić w takiej kolejności aby każda z kolejnych liczb sąsiadujących ze sobą również dawała największą możliwą wartość.

Jeżeli przykładowo w tablicy [7, 9, 10] kombinacja 97 daje większą wartość niż 79, to pozycje tych liczb należy zamienić w tablicy, aby utworzyć [9, 7, 10].

## 3. Podział programu

Program składa się z pięciu funkcji, których zastosowanie zostało opisane poniżej:

- **void wczytanie\_danych\_wejscowych(void)**
- **int main()**
- **int czy\_druga\_liczba\_jest\_bardziej\_znaczaca(int pierwsza\_liczba, int druga\_liczba)**
- **void zapis\_wyniku(void)**
- **void wygenerowanie\_danych\_wejscowych**

### 3.1. Odczyt danych wejściowych

Wszystkie liczby, z których powinna zostać utworzona największa możliwa liczba powinny zostać zapisane w pliku tekstowym **"dane\_wejscowe.txt"** lub wygenerowane losowo do tablicy z funkcji **void wygenerowanie\_danych\_wejscowych**

Za odczyt danych wejściowych odpowiada funkcja **void wczytanie\_danych\_wejscowych(void)** lub funkcja **void wygenerowanie\_danych\_wejscowych(void)**. Funkcja wczytująca dane z pliku tekstowego sprawdza ile liczb jest zapisanych w pliku tekstowym **dane\_wejscowe.txt**, a następnie dokonuje alokacji miejsca w tablicy danych, której ilość elementów odpowiada ilości liczb z pliku tekstowego. Następnie funkcja ta odczytuje kolejno liczby z pliku tekstowego i umieszcza je w tablicy danych. Funkcja generująca losowe liczby alokuje miejsce do tablicy danych, której ilość elementów odpowiada ilości wygenerowanych liczb

### 3.2. Sortowanie

Za sortowanie liczb w tablicy danych odpowiadają funkcje **int main()** oraz **int czy\_druga\_liczba\_jest\_bardziej\_znaczaca(int pierwsza\_liczba, int druga\_liczba)**.

Za sprawdzenie która kombinacja dwóch sąsiadujących ze sobą liczb daje większą wartość odpowiada funkcja **int czy\_druga\_liczba\_jest\_bardziej\_znaczaca(int pierwsza\_liczba, int druga\_liczba)**, do której przesyłane są kolejno po dwie sąsiadujące ze sobą liczby z tablicy danych. Powyższa funkcja zwraca wartość:

- 0, kiedy kombinacja <pierwsza\_liczba><druga\_liczba> daje większą wartość niż <druga\_liczba><pierwsza\_liczba>, co oznacza że druga liczba nie jest bardziej znacząca.
- 1, w odwrotnej sytuacji, kiedy kombinacja <druga\_liczba><pierwsza\_liczba> daje większą wartość niż <pierwsza\_liczba><druga\_liczba>, czyli w sytuacji kiedy druga liczba jest bardziej znacząca.

Na podstawie zwróconego wyniku funkcja **int main()** podejmuje decyzje czy pozycje dwóch sąsiadujących liczb należy zamienić. Jeśli wynikiem działania funkcji **int czy\_druga\_liczba\_jest\_bardziej\_znacząca(int pierwsza\_liczba, int druga\_liczba)** jest wartość:

- 0, to dwie sąsiadujące ze sobą liczby należy zostawić na aktualnych pozycjach, gdyż druga z sąsiadujących liczb nie jest bardziej znacząca,
- 1, to pozycje dwóch sąsiadujących ze sobą liczb należy zamienić, gdyż druga liczba jest bardziej znacząca.

Funkcja **int main()** wywołuje w pętli funkcję **int czy\_druga\_liczba\_jest\_bardziej\_znacząca(int pierwsza\_liczba, int druga\_liczba)**, przekazując do niej kolejno następne sąsiadujące ze sobą liczby z tablicy danych, do momentu kiedy żadna para sąsiadujących ze sobą liczb w tablicy nie będzie wymagała zamiany pozycji, co oznacza że wszystkie liczby ułożono już na pozycjach, które w wyniku dają największą możliwą liczbę.

### 3.3. Zapis wyniku

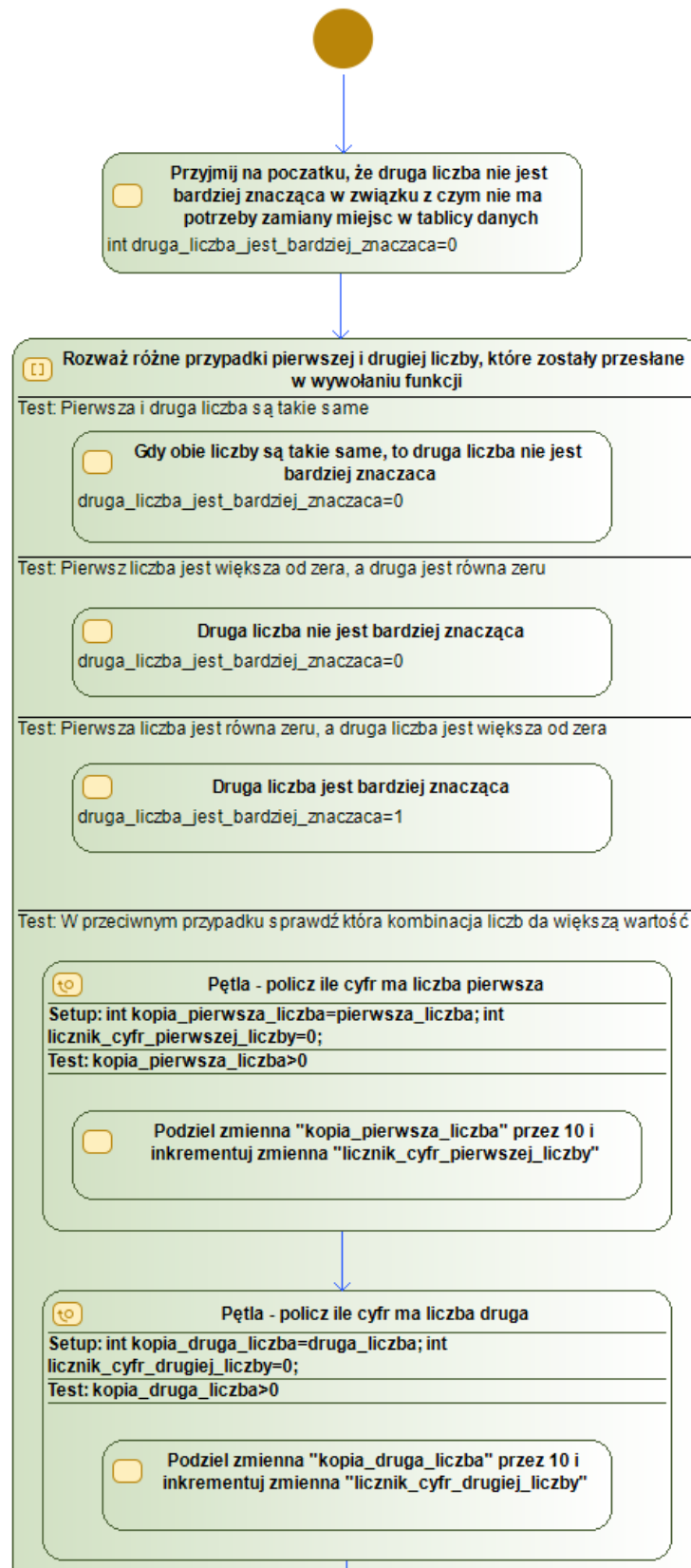
Funkcja **void zapis\_wyniku(void)** jest odpowiedzialna za zapisanie liczb z posortowanej tablicy danych do pliku wyjściowego **wynik.txt**.

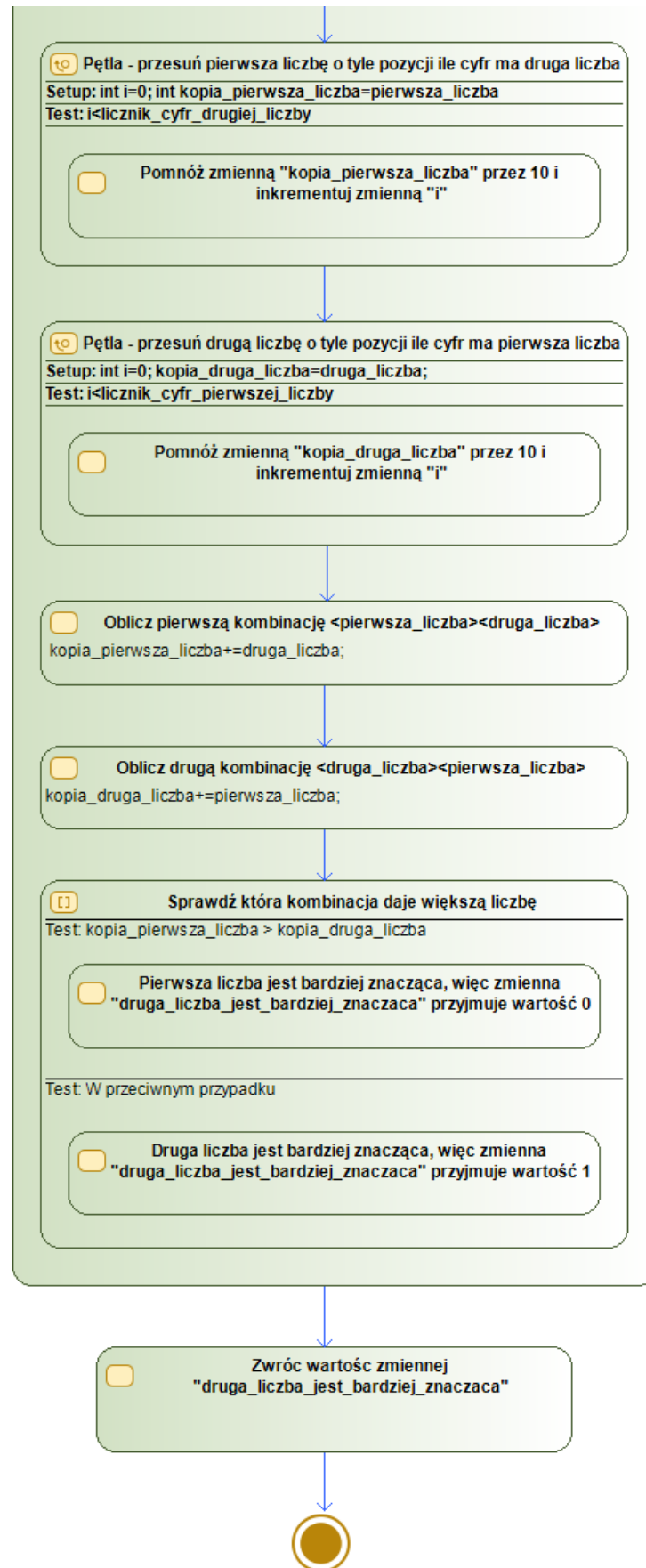
### 3.4. Wygenerowanie danych wejściowych

Funkcja **void wczytanie\_danych\_wejsciowych** generuje losowe liczby do tablicy od 0 do 100, które będą służyły do przeprowadzenia złożoności obliczeniowej algorytmu.

## 4. Schemat blokowy głównego algorytmu

Funkcja `int czy_druga_liczba_jest_bardziej_znaczaca (int pierwsza_liczba, int druga_liczba)`





## 5. Pseudokod głównego algorytmu

### Krok 1:

Zmienna "druga\_liczba\_jest\_bardziej\_znaczaca" może przyjąć następujące wartości:

0 – druga liczba nie jest bardziej znacząca, w związku z czym nie ma potrzeby zamiany miejsc liczb w tablicy danych.

1- druga liczba jest bardziej znacząca, w związku z czym dwie sąsiadujące ze sobą liczby powinny zostać zamienione w tablicy danych.

### Krok 2:

Rozważ wszystkie przypadki pierwszej i drugiej liczby, które zostały przesłane z pliku tekstowego lub z tablicy wypełnionej losowymi liczbami.

### Krok 3:

Policz ile cyfr ma pierwsza i druga liczba dzieląc je przez dziesięć.

### Krok 4:

Przesuń pierwszą liczbę o tyle cyfr ile ma druga liczba oraz przesuń drugą liczbę o tyle cyfr ile ma pierwsza liczba.

### Krok 5:

Oblicz pierwszą kombinację <pierwsza\_liczba><druga\_liczba> oraz drugą kombinację <druga\_liczba><pierwsza\_liczba>.

### Krok 6:

Sprawdź, która kombinacja daje większą liczbę.

### Krok 7:

Zwróć wartość zmiennej "druga\_liczba\_jest\_bardziej\_znaczaca".

## 6. Kod źródłowy głównego algorytmu

```
/* Funkcja sprawdza jaka kombinacja dwóch podanych liczb pozwoli na stworzenie wiekszej  
liczby poprzez ich polaczenie wedlug schematu:
```

```
- kombinacja 1: <pierwsza_liczba> <druga_liczba>
```

```
- kombinacja 2: <druga_liczba> <pierwsza_liczba>
```

```
Jezeli nowa liczba utworzona z kombinacji 2 daje wieksza wartosc niz z kombinacji 1, to  
funkcja zwraca wartosc 1, w przeciwnym przypadku zwraca wartosc 0 */
```

```
int czy_druga_liczba_jest_bardziej_znaczaca(int pierwsza_liczba, int druga_liczba)  
{
```

```
    /*Zmienna okresla czy druga liczba jest bardziej znaczaca, czy liczba z kombinacji  
    drugiej jest wieksza od liczby z kombinacji pierwszej, jest to zmienna zwraca przez funkcje*/  
    int druga_liczba_jest_bardziej_znaczaca=0;
```

```
    //Wyswietla mozliwe kombinacje
```

```
    cout<<endl;
```

```
    cout<<"Pozycja 1 <pierwszaliczba> <drugaliczba> "<<pierwsza_liczba<<druga_liczba<<endl;
```

```
    cout<<"Pozycja 2 <drugaliczba> <pierwszaliczba> "<<druga_liczba<<pierwsza_liczba<<endl;
```

```
    //Sprawdza typowe przypadki
```

```
    if(pierwsza_liczba==druga_liczba)
```

```
    {
```

```
        //Jesli obie liczby maja te sama wartosc to druga liczba nie jest bardziej znaczaca
```

```
        druga_liczba_jest_bardziej_znaczaca=0;
```

```
    }
```

```

else if(pierwsza_liczba>0&&druga_liczba==0)
{
    /*Jesli pierwsza liczba jest wieksza od 0, a druga liczba jest rowna 0, to druga liczba nie jest bardziej
znaczaca*/
    druga_liczba_jest_bardziej_znaczaca=0;
}

else if(pierwsza_liczba==0&&druga_liczba>0)
{
    /*Jesli pierwsza liczba jest rowna 0, a druga liczba jest wieksza od 0, to druga liczba jest bardziej
znaczaca*/
    druga_liczba_jest_bardziej_znaczaca=1;
}

else
{
    int licznik_cyfr_pierwszej_liczby=0;
    int licznik_cyfr_drugiej_liczby=0;
    int kopia_pierwsza_liczba=pierwsza_liczba;
    int kopia_druga_liczba=druga_liczba;

    //Liczy ile cyfr ma pierwsza liczba
    while(kopia_pierwsza_liczba>0)
    {
        kopia_pierwsza_liczba /=10;
        licznik_cyfr_pierwszej_liczby++;
    }

    //Liczy ile cyfr ma druga liczba
    while(kopia_druga_liczba>0)
    {
        kopia_druga_liczba /=10;
        licznik_cyfr_drugiej_liczby++;
    }

    //Wczytuje pierwsza i druga liczbe do kopii po przeliczeniu cyfr
    kopia_pierwsza_liczba=pierwsza_liczba;
    kopia_druga_liczba=druga_liczba;

    //Przesuwa pierwsza liczbe o tyle pozycji ile cyfr ma druga liczba
    for(int i=0; i<licznik_cyfr_drugiej_liczby; i++)
    {
        kopia_pierwsza_liczba *=10;
    }

    //Przesuwa druga liczbe o tyle pozycji ile cyfr ma pierwsza liczba
    for(int i=0; i<licznik_cyfr_pierwszej_liczby; i++)
    {
        kopia_druga_liczba *=10;
    }
}

```

```

//Wylicza kombinacje 1
kopia_pierwsza_liczba+=druga_liczba;

//Wylicz kombinacje 2
kopia_druga_liczba+=pierwsza_liczba;

//Jesli liczba z kombinacji pierwszej jest wieksza od liczby z kombinacji drugiej
if(kopia_pierwsza_liczba>kopia_druga_liczba)
{
    //To druga liczba nie jest bardziej znacząca
    druga_liczba_jest_bardziej_znacząca=0;
}
else
{
    //Druga liczba jest bardziej znacząca
    druga_liczba_jest_bardziej_znacząca=1;
}

//Wyswietla wynik
cout<<"Druga liczba jest bardziej znacząca: "<<druga_liczba_jest_bardziej_znacząca<<endl;
}

// zwroc wynik
return druga_liczba_jest_bardziej_znacząca;
}

```

## 7. Znane ograniczenia programu

Program wczytuje liczby z pliku wejściowego do zmiennych lub generuje liczby losowe oraz oczekuje tylko liczb całkowitych dodatnich lub zera.

## 8. Uruchomienie programu i sprawdzenie wyników

Przykłady testów wykonanych wewnątrz programu:

### Test 1:

Wejście: wczytanie liczb [9 10 81]

Wyjście: liczba po sklejeniu 98110

### Test 2:

Wejście: wczytanie liczb [65 9 88 7]

Wyjście: liczb po sklejeniu 988765

### Test 3:

Wejście: wczytanie liczb [54 8 7 2 10]

Wyjście: liczb po sklejeniu 8754210



#### Test 4:

Wejście: wczytanie liczb [1 98 55 76 9 43 2 ]

Wyjście: liczba po sklejeniu 99876554321

### 9. Złożoność czasowa algorytmu

Algorytm ma złożoność czasową  $O(n^2)$ .



### 10. Podsumowanie

Algorytm ma za zadanie sklejać całość z podanych liczb w największą możliwą. Program ten wykorzystuje do tego dzielenie zwykłe przez dziesięć aby porównać cyfry pierwsze oraz sortowanie bąbelkowego do porównania ze sobą kolejnych liczb w tablicy.